

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) A method of controlling object oriented computer software applications in a computing device having an operating system, comprising:

providing at least onea plurality of application ~~wrapper~~wrappers that run independently of said operating system, each of the application wrappers providing an object oriented method of controlling a single object oriented software application;

assigning at leasta single one of a plurality of object oriented software ~~application~~applications to each said at least one application wrapper;

providing a stable interface between [[an]]said operating system and each said application wrapper so that changes may be made to either without affecting the functionality of the other;

generating a display of each said application wrapper for a computer user;

accessing data contained in each said software application in a modular format in response to computer user inputs to each said application wrapper; and

managing at least one of the operation, execution, performance, security, functions and resource usage of each said software application in a modular format in response to computer user inputs to each said application wrapper

~~controlling the functionality of each said software application in a modular format in response to computer user inputs to each said application wrapper.~~

2. (original) The method of controlling object oriented computer software applications of claim 1, further comprising the step of starting at least one application wrapper upon power up of the computer system.

3. (original) The method of controlling object oriented computer software

applications of claim 1, further comprising the step of providing a link between said operating system and each said software application wherein the operating system can ascertain if an object is one that each said software application can interpret.

4. (original) The method of controlling object oriented computer software applications of claim 1, further comprising the step of providing a link between said operating system and each said software application for notifying each said software application of changes in the operating system.

5. (original) The method of controlling object oriented computer software applications of claim 4, wherein said link notifies each said software application of a pending power down of the computer system.

6. (original) The method of controlling object oriented computer software applications of claim 4, where in said link notifies each said software application of a power up of the computer system.

7. (original) The method of controlling object oriented computer software applications of claim 1, further comprising the step of generating at least one identifying means on a display for assisting the computer user in opening and gaining access to each said application wrapper.

8. (original) The method of controlling object oriented computer software applications of claim 7, where in said identifying means is displayed on said display by an animated icon.

9. (original) The method of controlling object oriented computer software applications of claim 7, wherein said identifying means is selected from the group including sound, computer animation, static images and video images.

10. (original) The method of controlling object oriented computer software applications of claim 1, further comprising the step of generating a display of at least one help file of said software applications in response to computer user inputs to each said application wrapper.

11. (original) The method of controlling object oriented computer software applications of claim 1, where in said date contained within each said software application comprises configuration settings.

12. (currently amended) The method of controlling object oriented computer software applications of claim 1, further comprising the step of starting each said software application in response to computer user inputs to each said plurality of application wrappers.

13. (currently amended) The method of controlling object oriented computer software applications of claim 1, further comprising the step of exiting each said software application in response to computer user inputs to each said plurality of application wrappers.

14. (currently amended) The method of controlling object oriented computer software applications of claim 1, further comprising the step of communicating with other software applications on the computer system in response to computer user inputs to each said plurality of application wrappers.

15. (currently amended) The method of controlling object oriented computer software applications of claim 1, further comprising the step of deleting said software applications and all of their dependent objects from the computer system in response to computer user inputs to each said plurality of application wrappers.

16. (currently amended) The method of controlling object oriented computer

software applications of claim 1, further comprising the step of linking a plurality of said software applications together so that an object is visible within each of the linked plurality of said software ~~applications~~ application without affecting the functionality of any software application.

17. (original) The method of controlling object oriented computer software applications of claim 1, further comprising the step of associating each application wrapper stored within the computer system to a common application wrapper database.

18. (original) The method of controlling object oriented computer software applications of claim 17, wherein said common application wrapper database is accessible by the computer user through an input device to change configuration settings and to access select data files associated with each application wrapper.

19. (original) The method of controlling object oriented computer software applications of claim 18, wherein said configuration settings and data files include at least one database file selected from the group comprising a computer user's personal profile, a specific software application's bubble help, a plurality of settings to control an application wrapper's icon appearance and behavior, an option to launch the software application in a secure process, and a limit on said software applications allowable memory usage and priority.

20-36. (canceled)

37. (currently amended) A method of controlling object oriented computer software applications in a computing device having an operating system, comprising:

providing at least one a plurality of application ~~wrapper~~ wrappers that run independently of said operating system, each of the application wrappers providing an object oriented method of controlling a single object oriented software application;

assigning at least a single one of a plurality of object oriented software

applicationapplications to each said application wrapper;

providing a stable interface between [[an]]said operating system and each said application wrapper so that changes may be made to either without affecting the functionality of the other;

generating a display of each said application wrapper for a computer user; and

managing at least one of the operation, execution, performance, security, functions and resource usage of each said software application in a modular format in response to computer user inputs to each said application wrapper

~~controlling the functionality of each said software application in a modular format in response to computer user inputs to each said application wrapper.~~

38. (original) The method of controlling object orientation computer software applications of claim 37, further comprising the step of starting at least one application wrapper upon power up of the computer system.

39. (original) The method of controlling object oriented computer software applications of claim 37, further comprising the step of providing a link between said operating system and each said software application wherein the operating system can ascertain if an object is one that each said software application can interpret.

40. (original) The method of controlling object oriented computer software applications of claim 37, further comprising the step of providing a link between said operating system and each said software application for notifying each said software application of changes in the operating system.

41. (original) The method of controlling object oriented computer software applications of claim 40, wherein said link notifies each said software application of a pending power down of the computer system.

42. (original) The method of controlling object oriented computer software

applications of claim 40, wherein said link notifies each said software application of a power up of the computer system.

43. (original) The method of controlling object oriented computer software applications of claim 37, further comprising the step of generating at least one identifying means on a display for assisting the computer user in opening and gaining access to each said application wrapper.

44. (original) The method of controlling object oriented computer software applications of claim 43, wherein said identifying means is displayed on said display by an animated icon.

45. (original) The method of controlling object oriented computer software applications of claim 43, wherein said identifying means is selected from the group including sound, computer animation, static images and video images.

46. (currently amended) The method of controlling object oriented computer software applications of claim 37, further comprising the step of generating a display of at least one help file of said at least one of said software applications in response to computer user inputs to each one of said application wrapperwrappers.

47. (currently amended) The method of controlling object oriented computer software applications of claim 37, further comprising the step of starting each said software application in response to computer user inputs to each said application wrapperwrapper.

48. (currently amended) The method of controlling object oriented computer software applications of claim 37, further comprising the step of exiting each said software application in response to computer user inputs to each said application wrapperwrapper.

49. (original) The method of controlling object oriented computer software applications of claim 37, further comprising the step of communicating with other software applications on the computer system in response to computer user inputs to said application wrappers.

50. (currently amended) The method of controlling object oriented computer software applications of claim 37, further comprising the step of deleting each said software application and all of their its dependent objects from the computer system in response to computer user inputs to each said application wrapper.

51. (currently amended) The method of controlling object oriented computer software applications of claim 37, further comprising the step of linking a plurality of said software applications together so that an object is visible within each said plurality of software application without affecting the functionality of any software application.

52. (original) The method of controlling object oriented computer software applications of claim 37, further comprising the step of associating each application wrapper stored within the computer system to a common application wrapper database.

53. (original) The method of controlling object oriented computer software applications of claim 52, wherein said common application wrapper database is accessible by the computer user through an input device to change configuration settings and to access select data files associated with each application wrapper.

54. (currently amended) The method of controlling object oriented computer software applications of claim 53, wherein said configuration settings and data files may include at least one database file selected from the group comprising a computer user's personal file, a specific software application's bubble help, a plurality of settings to control an application wrapper wrapper's icon appearance and behavior, an option to launch the software application

in a secure process, and a limit on said software applications allowable memory usage and priority.

55-67. (canceled)

AMENDMENTS TO THE DRAWINGS

The attached sheets of drawings are provided to replace the originally filed informal drawings with formal drawings. There are no amendments made to the drawings. The sheets, which include Figures 1-3, replace the original sheets including Figures 1-3.

Attachment: 3 Replacement Sheets